

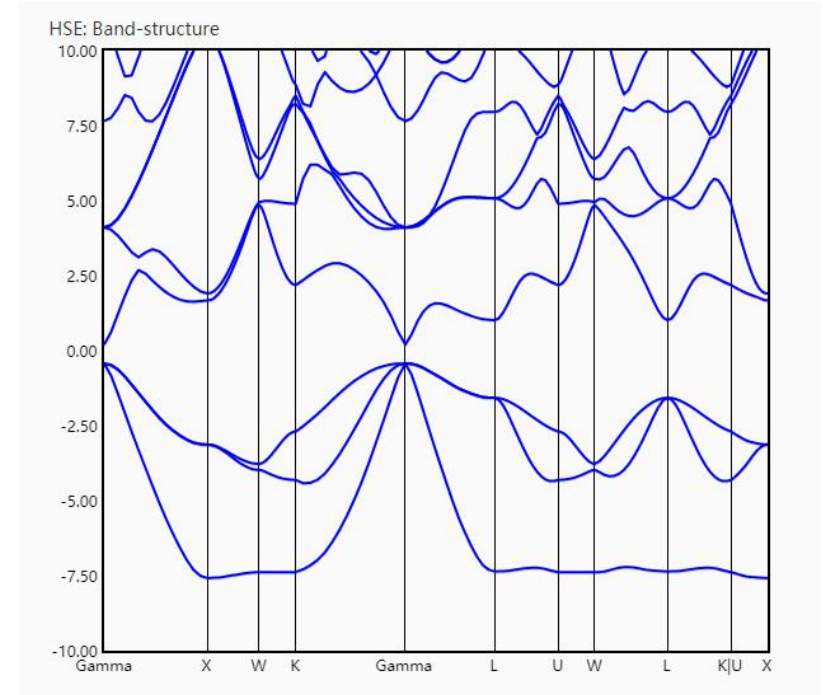
8-能带、投影能带、态密度、投影  
态密度计算

# 能带和态密度简介

回顾一下基本概念：

1. K 点是什么？
2. 能带图的横坐标代表什么？
3. 能带图的纵坐标代表什么？
4. 横坐标上的字母表示什么意思？

1. 什么是电子态密度？
2. 电子态密度图和能带图有什么关联？



典型的GaAs HSE 计算的能带图

# 能带计算

为了画出能带，我们需要计算出什么？

需要计算：一系列能量本征值， $E_n(\mathbf{k})$

$\mathbf{K}$ 表示第 $k$ 个 $\mathbf{K}$ 点

$n$ 表示第 $n$ 个能级

在之前的计算中，我们已经可以得到类似的结果了

以下是一个典型的**REPORT** 文件

```
iisllda,kpt= 1, 42 0.233640 0.404676 -0.143074 kpoint in xyz unit
err of each states, A.U
0.404169E-04 0.372011E-04 0.389562E-04 0.364137E-04 0.358264E-04
0.164417E-04 0.142738E-04 0.121970E-04 0.119284E-04 0.116945E-04
0.962264E-05 0.110676E-04 0.142244E-04 0.137176E-04 0.128813E-04
0.144689E-04 0.200549E-04 0.451714E-04 0.878274E-04
eigen energies, in eV
-13.775331 -13.668385 -13.645556 -13.632407 -13.612172
-9.007644 -3.754183 -0.347898 1.030192 7.606015
8.726208 10.399985 12.149619 13.714626 15.949533
17.234128 18.455220 20.812606 21.407616
*****
```

第42个  $\mathbf{K}$  点

$\mathbf{K}$ 点的坐标

一系列能量本征值

# 能带计算

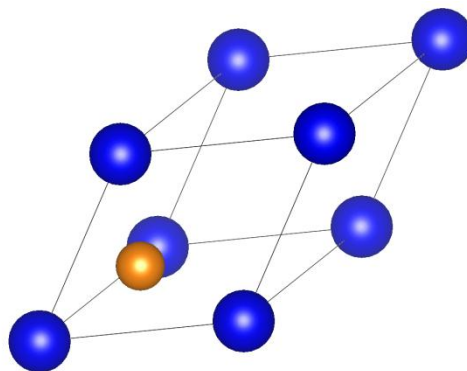
## 能带计算步骤:

- 首先进行**SCF**计算
- 将**SCF**计算得到的**OUT.VR**拷贝成**IN.VR**，如果开了自旋还需把**OUT.VR\_2**拷贝成**IN.VR\_2**
- 准备高对称点路径文件
- 进行**NONSCF**计算
- 数据处理，得到能带图和画能带的文件

# 能带计算

首先进行SCF计算:

输入文件包括: **etot.input**, **atom.config**, **\*.UPF**



**GaAs**

```
4 1
IN.ATOM = atom.config
JOB = SCF
MP_N123= 7 7 7 0 0 0
in.psp1 = Ga.nc.lda.UPF
in.psp2 = As.nc.lda.UPF
```

**etot.input**

```
2
Lattice vector
3.9974000454 0.0000000000 0.0000000000
1.9987000227 3.4618499884 0.0000000000
1.9987000227 1.1539499961 3.2638634697
Position, move_x, move_y, move_z
33 0.250000000000 0.250000000000 0.250000000000 1 1 1
31 0.000000000000 0.000000000000 0.000000000000 1 1 1
```

**atom.config**

# 能带计算

进行NONSCF计算:

- SCF计算得到的OUT.VR拷贝成IN.VR，如果开了自旋还需把OUT.VR\_2拷贝成IN.VR\_2
- 准备高对称点路径文件gen.kpt，格式如下，然后使用split\_kp.x生成PWmat可以读取的K点路径，使用方法，split\_kp.x < gen.kpt

```
Band
15
0.500 0.250 0.750
0.500 0.500 0.500
15
0.500 0.500 0.500
0.000 0.000 0.000
15
0.000 0.000 0.000
0.500 0.000 0.500
15
0.500 0.000 0.500
0.500 0.250 0.750
15
0.500 0.250 0.750
0.375 0.375 0.750
```

gen.kpt

```
5
2 0
0.500000 0.250000 0.750000 1.000000
0.500000 0.265625 0.734375 1.000000
0.500000 0.281250 0.718750 1.000000
0.500000 0.296875 0.703125 1.000000
0.500000 0.312500 0.687500 1.000000
0.500000 0.328125 0.671875 1.000000
0.500000 0.343750 0.656250 1.000000
0.500000 0.359375 0.640625 1.000000
0.500000 0.375000 0.625000 1.000000
0.500000 0.390625 0.609375 1.000000
0.500000 0.406250 0.593750 1.000000
0.500000 0.421875 0.578125 1.000000
0.500000 0.437500 0.562500 1.000000
0.500000 0.453125 0.546875 1.000000
0.500000 0.468750 0.531250 1.000000
0.500000 0.484375 0.515625 1.000000
0.500000 0.500000 0.500000 1.000000
0.500000 0.500000 0.500000 1.000000
0.468750 0.468750 0.468750 1.000000
. . .
0.390625 0.359375 0.750000 1.000000
0.382812 0.367188 0.750000 1.000000
0.375000 0.375000 0.750000 1.000000
```

IN.KPT

如何选取高对称点:

- 1.一般而言，我们需要选取高对称的路径
- 2.为了选取这些高对称路径，我们可以使用MaterialsStudio生成，也可以使用spglib等库生成
- 3.不同软件生成的高对称路径会有细微差别，有的会多几条路径等

# 能带计算

进行NONSCF计算:

- 修改`etot.input`文件用于NONSCF计算, 设置“`JOB=NONSCF`”, 另外读取`IN.KPT`和`IN.VR`文件, **特别注意**, 这里读取的是`IN.KPT`里面的K点, 所以需要删除`MP_N123`参数。

```
1
IN.ATOM = atom.config
JOB = NONSCF
IN.KPT=T
in.psp1 = Ga.nc.lda.UPF
in.psp2 = As.nc.lda.UPF
IN.VR=T
```

`etot.input`

```
2
Lattice vector
3.9974000454      0.0000000000      0.0000000000
1.9987000227      3.4618499884      0.0000000000
1.9987000227      1.1539499961      3.2638634697
Position, move_x, move_y, move_z
33  0.250000000000  0.250000000000  0.250000000000  1  1  1
31  0.000000000000  0.000000000000  0.000000000000  1  1  1
```

`atom.config`

- 计算完成后, 在计算目录下运行`plot_band_structure.x`, 运行完毕后会得到下列文件

```
bandstructure_1.txt  bandstructure.eps  bandstructure.pdf  bandstructure.png
```

能带数据文件

eps格式的能带图

pdf格式的能带图

png格式的能带图

- **特别注意**, 如果开了自旋, 还有`bandstructure_2.txt`文件, **1和2**分别为不同自旋的能带数据

# 能带计算

如果在能带图的横坐标上标注高对称点？

bandstructure\_1.txt文件

K点距离      本征值

```
0.0000 -14.6589281046
0.0378 -14.6582222146
0.0756 -14.6570579646
0.1134 -14.6549586046
0.1512 -14.6523252646
0.1890 -14.6494996146
0.2268 -14.6466587246
0.2647 -14.6441332946
0.3025 -14.6421362546
0.3403 -14.6406647546
0.3781 -14.6399344246
0.4159 -14.6396924446
0.4159 -14.6396946346
0.4622 -14.6380164446
0.5085 -14.6334375246
0.5548 -14.6255874746
0.6011 -14.6152103546
```

...

```
1.7273 -14.6615385946
1.7540 -14.6600333446
1.7807 -14.6589976146
1.8075 -14.6589273846
1.8075 -14.6589243046
1.8264 -14.6587248546
1.8453 -14.6588040246
1.8642 -14.6589902646
1.8831 -14.6591410346
1.9020 -14.6593726046
1.9209 -14.6594599146
1.9398 -14.6597657246
1.9587 -14.6599000346
1.9776 -14.6600333446
1.9965 -14.6602583446
2.0154 -14.6603041246

0.0000 -14.5051357746
0.0378 -14.5051989346
0.0756 -14.5063416946
0.1134 -14.5077605646
0.1512 -14.5093944946
```

```
Band
15
0.500 0.250 0.750
0.500 0.500 0.500
15
0.500 0.500 0.500
0.000 0.000 0.000
15
0.000 0.000 0.000
0.500 0.000 0.500
15
0.500 0.000 0.500
0.500 0.250 0.750
15
0.500 0.250 0.750
0.375 0.375 0.750
```

不同带之间用空格分开

# HSE06计算能带

从REPORT 中读取带隙大小的思路

1. 读取所有Eigen energies
2. 判断是否有能带穿过费米能级：找到每条能带的最大最小的eigen energy
3. 如果 $\text{Max} > \text{Fermi} \ \&\& \ \text{Min} < \text{Fermi}$  则穿过了费米能级，体系为导体
4. 如果不为导体：找到比Fermi Level 大的最小的本征能量，找到比费米能级小的最大的本征能量
5. 把上述两项做差，即为带隙

可以试着写一个小程序

# HSE06计算能带

与LDA 计算类似，分别计算SCF 和NONSCF 两个步骤  
在输入文件中加上XCFUNCTIONAL = HSE

```
4 1
  IN.ATOM  =  atom.config
  JOB      =  SCF
MP_N123= 7 7 7 0 0 0
in.psp1 = Ga.nc.lda.UPF
in.psp2 = As.nc.lda.UPF
XCFUNCTIONAL=HSE
```

SCF 输入文件

```
4 1
  IN.ATOM  =  atom.config
  JOB      =  NONSCF
IN.KPT=T
in.psp1 = Ga.nc.lda.UPF
in.psp2 = As.nc.lda.UPF
XCFUNCTIONAL=HSE
IN.VR=T
```

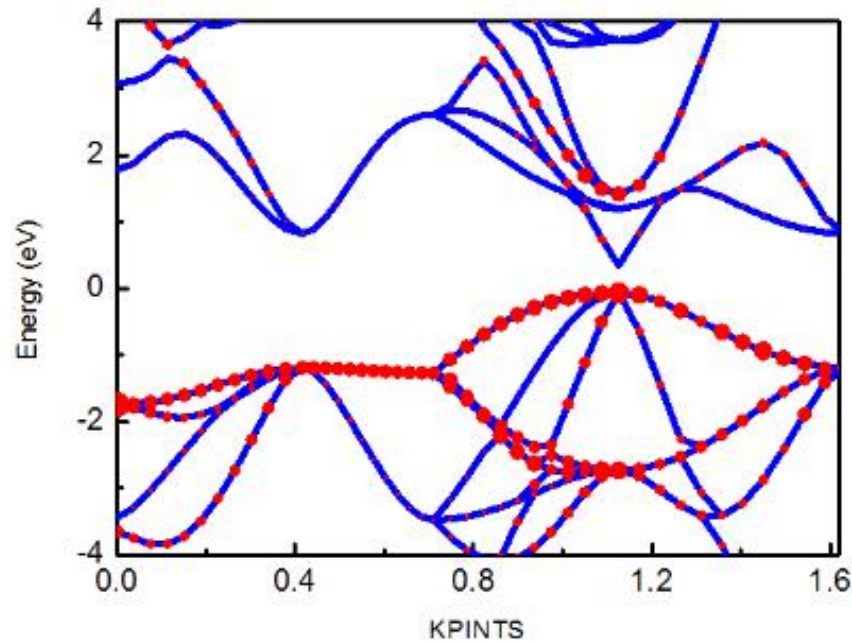
NONSCF 输入文件

注意：

1. HSE 不能使用K点并行
2. HSE 计算NONSCF 的时候，除了需要读取IN.VR 以外，还需要读取SCF 计算生成的OUT.HSEWR\* 文件

# 投影能带

特定轨道对每根能带的贡献



# 投影能带

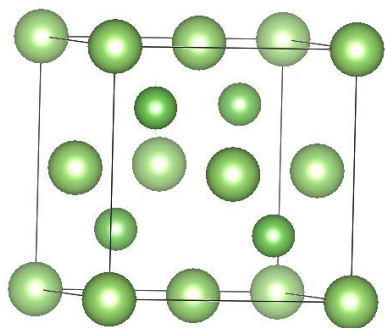
## 投影能带计算步骤:

- 首先进行**SCF**计算
- 将**SCF**计算得到的**OUT.VR**拷贝成**IN.VR**，如果开了自旋还需把**OUT.VR\_2**拷贝成**IN.VR\_2**
- 准备高对称点路径文件
- 进行**NONSCF**计算
- 读入**NONSCF**的**IN.KPT**进行”**JOB=DOS**”计算
- 数据处理，提取画投影能带所需要的数据，在计算目录下运行:**plot\_fatband\_structure.x**

# 投影能带

首先进行SCF计算:

输入文件包括: **etot.input**, **atom.config**, **\*.UPF**



```
4 1
IN.ATOM = atom.config
JOB = SCF
IN.PSP1 = Ga.SG15.PBE.UPF
IN.PSP2 = As.SG15.PBE.UPF
XCFUNCTIONAL = PBE
MP_N123 = 6 6 6 0 0 0
```

**etot.input**

```
8
LATTICE
5.65320015 0.00000000 0.00000000
0.00000000 5.65320015 0.00000000
0.00000000 0.00000000 5.65320015
POSITION
33 0.25000000 0.25000000 0.25000000 1 1 1
33 0.75000000 0.75000000 0.25000000 1 1 1
33 0.75000000 0.25000000 0.75000000 1 1 1
33 0.25000000 0.75000000 0.75000000 1 1 1
31 0.00000000 0.00000000 0.00000000 1 1 1
31 0.00000000 0.50000000 0.50000000 1 1 1
31 0.50000000 0.00000000 0.50000000 1 1 1
31 0.50000000 0.50000000 0.00000000 1 1 1
```

**atom.config**

# 能带计算

进行NONSCF计算:

- SCF计算得到的OUT.VR拷贝成IN.VR，如果开了自旋还需把OUT.VR\_2拷贝成IN.VR\_2
- 准备高对称点路径文件gen.kpt，格式如下，然后使用split\_kp.x生成PWmat可以读取的K点路径，使用方法，split\_kp.x < gen.kpt

```
Band
10
0.500 0.000 0.000
0.500 0.500 0.500
10
0.500 0.500 0.500
0.500 0.500 0.000
10
0.500 0.500 0.000
0.000 0.000 0.000
10
0.000 0.000 0.000
0.500 0.500 0.500
```

gen.kpt

```
2
0
0.500000 0.000000 0.000000 1.000000
0.500000 0.045455 0.045455 1.000000
0.500000 0.090909 0.090909 1.000000
0.500000 0.136364 0.136364 1.000000
0.500000 0.181818 0.181818 1.000000
0.500000 0.227273 0.227273 1.000000
0.500000 0.272727 0.272727 1.000000
0.500000 0.318182 0.318182 1.000000
0.500000 0.363636 0.363636 1.000000
0.500000 0.409091 0.409091 1.000000
0.500000 0.454545 0.454545 1.000000
0.500000 0.500000 0.500000 1.000000
0.500000 0.500000 0.500000 1.000000
. . .
0.454545 0.454545 0.454545 1.000000
0.500000 0.500000 0.500000 1.000000
```

IN.KPT

# 投影能带

进行NONSCF计算:

- 修改`etot.input`文件用于NONSCF计算, 设置“`JOB=NONSCF`”, 另外读取`IN.KPT`和`IN.VR`文件, **特别注意**, 这里读取的是`IN.KPT`里面的K点, 所以需要删除`MP_N123`参数。

```
4 1
IN.ATOM = atom.config
JOB = NONSCF
IN.PSP1 = Ga.SG15.PBE.UPF
IN.PSP2 = As.SG15.PBE.UPF
XCFUNCTIONAL = PBE
IN.VR = T
IN.KPT = T
```

**etot.input**

```
8
LATTICE
5.65320015 0.00000000 0.00000000
0.00000000 5.65320015 0.00000000
0.00000000 0.00000000 5.65320015
POSITION
33 0.25000000 0.25000000 0.25000000 1 1 1
33 0.75000000 0.75000000 0.25000000 1 1 1
33 0.75000000 0.25000000 0.75000000 1 1 1
33 0.25000000 0.75000000 0.75000000 1 1 1
31 0.00000000 0.00000000 0.00000000 1 1 1
31 0.00000000 0.50000000 0.50000000 1 1 1
31 0.50000000 0.00000000 0.50000000 1 1 1
31 0.50000000 0.50000000 0.00000000 1 1 1
```

**atom.config**

# 投影能带

进行DOS计算:

- 修改`etot.input`文件用于NONSCF计算, 设置“JOB=DOS”, 另外读取`IN.KPT`和`IN.WG`文件, 特别注意, 这里读取的是`IN.KPT`里面的K点, 与前面NONSCF读取的`IN.KPT`要完全相同。

```
4 1
IN.ATOM = atom.config
JOB = DOS
IN.PSP1 = Ga.SG15.PBE.UPF
IN.PSP2 = As.SG15.PBE.UPF
XCFUNCTIONAL = PBE
IN.WG = T
IN.KPT = T
```

`etot.input`

```
8
LATTICE
5.65320015 0.00000000 0.00000000
0.00000000 5.65320015 0.00000000
0.00000000 0.00000000 5.65320015
POSITION
33 0.25000000 0.25000000 0.25000000 1 1 1
33 0.75000000 0.75000000 0.25000000 1 1 1
33 0.75000000 0.25000000 0.75000000 1 1 1
33 0.25000000 0.75000000 0.75000000 1 1 1
31 0.00000000 0.00000000 0.00000000 1 1 1
31 0.00000000 0.50000000 0.50000000 1 1 1
31 0.50000000 0.00000000 0.50000000 1 1 1
31 0.50000000 0.50000000 0.00000000 1 1 1
```

`atom.config`

# 投影能带

- 计算完成后，在计算目录下运行`plot_fatband_structure.x`，运行完毕后会得到下列文件

```
fatbandstructure_1.txt
```

能带数据文件

- 特别注意，如果开了自旋，还有`bandstructure_2.txt`文件，1和2分别为不同自旋的能带数据

能带序号      K点距离      本征值

不同轨道贡献值

BAND	POINT	ENERGY	weight_tot	Ga-dz2	Ga-dxz	Ga-dyz	Ga-dxy	Ga-dx2-y2	Ga-s	Ga-pz	Ga-px	Ga-py	As-s	As-pz	As-px	As-py
0.00000E+00	1	-0.15253E+02	0.10000E+01	0.23964E+00	0.23562E-08	0.24520E-01	0.71897E+00	0.18880E-08	0.62039E-03	0.60059E-10	0.34727E-03	0.47003E-10	0.10425E-02	0.50905E-11	0.14862E-01	0.11732E-11
0.37808E-01		-0.15254E+02	0.10000E+01	0.91221E-02	0.46277E+00	0.29018E-02	0.39049E-02	0.46280E+00	0.12426E-04	0.11104E-01	0.12062E-03	0.11105E-01	0.35986E-01	0.54230E-04	0.65469E-04	0.54223E-04
0.75615E-01		-0.15254E+02	0.10000E+01	0.64403E-01	0.42188E+00	0.47226E-02	0.24599E-01	0.42191E-02	0.12871E-02	0.10486E-01	0.81964E-03	0.10486E-01	0.38672E-01	0.36643E-03	0.10140E-05	0.36644E-03
0.11342E+00		-0.15252E+02	0.10000E+01	0.15124E+00	0.35878E+00	0.11636E-01	0.64019E-01	0.35881E+00	0.13504E-02	0.80129E-02	0.95890E-04	0.80130E-02	0.35581E-01	0.85446E-03	0.74716E-03	0.85449E-03
0.15123E+00		-0.15249E+02	0.10000E+01	0.15846E+00	0.34710E+00	0.25823E-01	0.68507E-01	0.34708E+00	0.21490E-02	0.68308E-02	0.13819E-03	0.68308E-02	0.32211E-01	0.19285E-02	0.10153E-02	0.19283E-02
0.18904E+00		-0.15252E+02	0.10000E+01	0.22819E+00	0.12739E-01	0.39881E-01	0.68449E+00	0.12740E-01	0.50032E-04	0.19478E-03	0.12348E-01	0.19482E-03	0.40467E-02	0.15352E-03	0.48240E-02	0.15349E-03
0.22685E+00		-0.15245E+02	0.10000E+01	0.19624E+00	0.61185E-01	0.72494E-01	0.58824E+00	0.61206E-01	0.37151E-03	0.60195E-03	0.43437E-02	0.60198E-03	0.93347E-02	0.92410E-03	0.35358E-02	0.92405E-03
0.26465E+00		-0.15249E+02	0.10000E+01	0.35057E+00	0.20222E+00	0.24337E-01	0.18345E+00	0.20222E+00	0.54064E-02	0.40667E-02	0.27099E-03	0.40665E-02	0.19968E-01	0.73836E-03	0.19449E-02	0.73809E-03
0.30246E+00		-0.15248E+02	0.10000E+01	0.68724E+00	0.38391E-01	0.20985E-07	0.22894E+00	0.38424E-01	0.43041E-09	0.19345E-02	0.78203E-09	0.19358E-02	0.26263E-08	0.15689E-02	0.28027E-08	0.15684E-02
0.34027E+00		-0.15256E+02	0.10000E+01	0.68191E+00	0.41739E-01	0.87267E-08	0.22687E+00	0.41793E-01	0.57874E-10	0.37304E-03	0.20699E-09	0.37353E-03	0.28981E-09	0.34724E-02	0.17318E-08	0.34698E-02
0.37808E+00		-0.15248E+02	0.10000E+01	0.64013E+00	0.60769E-01	0.81628E-02	0.21514E+00	0.60756E-01	0.60421E-03	0.17180E-03	0.51704E-04	0.17182E-03	0.21735E-02	0.58884E-02	0.87079E-04	0.58866E-02
...																
0*3J808E-07		-0*J2524E+03	0*J0000E+07	0*8T88E-05	0*4E5J8E+00	0*58078E-05	0*38047E-05	0*4E5J8E+00	0*JS454E-04	0*JJ70E-07	0*75028E-03	0*JJ70E-07	0*3288E-07	0*24550E-04	0*824J2E-04	0*24555E-04
0*00000E+00		-0*J2523E+05	0*J0000E+07	0*538EJ8E+00	0*30245E-08	0*54258E-07	0*J7883E+00	0*5808E-08	0*E504JE-08	0*3582E-70	0*34J34E-03	0*88808E-70	0*J0458E-05	0*J0874E-70	0*J4865E-07	0*J7038E-77
...																
0*J6325E+07		-0*J2558E+05	0*J0000E+07	0*8E048E+00	0*5T888E-07	0*JJ7J7E-07	0*5502JE-07	0*J5E22E-07	0*25244E-08	0*2EJJ3E-04	0*732J3E-04	0*JJ7J8E-04	0*8J28E-08	0*3E407E-05	0*82J60E-03	0*JJ054E-05
0*J2889E+07		-0*J2548E+05	0*J0000E+07	0*88038E+00	0*7E28JE-07	0*J8072E-07	0*55J38E-05	0*J0270E-07	0*55345E-08	0*8077JE-04	0*5T80JE-04	0*78858E-04	0*67044E-08	0*JJ758E-05	0*78780E-05	0*7E613E-05
0*J2458E+07		-0*J2525E+05	0*J0000E+07	0*8E883E+00	0*75808E-07	0*J5T20E-07	0*30558E-05	0*48E40E-07	0*7E0J0E-08	0*73524E-03	0*25438E-04	0*34438E-04	0*82320E-08	0*34808E-05	0*884J3E-03	0*80J83E-03

# 投影能带

如何计算部分原子的轨道对能带的贡献

**natom**

**LATTICE**

**a11 a21 a31**

**a12 a22 a32**

**a13 a23 a33**

**POSITION**

**N x y z 0 0 0 1** **#the last column, if setting 1, including this atom**

**N x y z 0 0 0 1**

**N x y z 0 0 0 1**

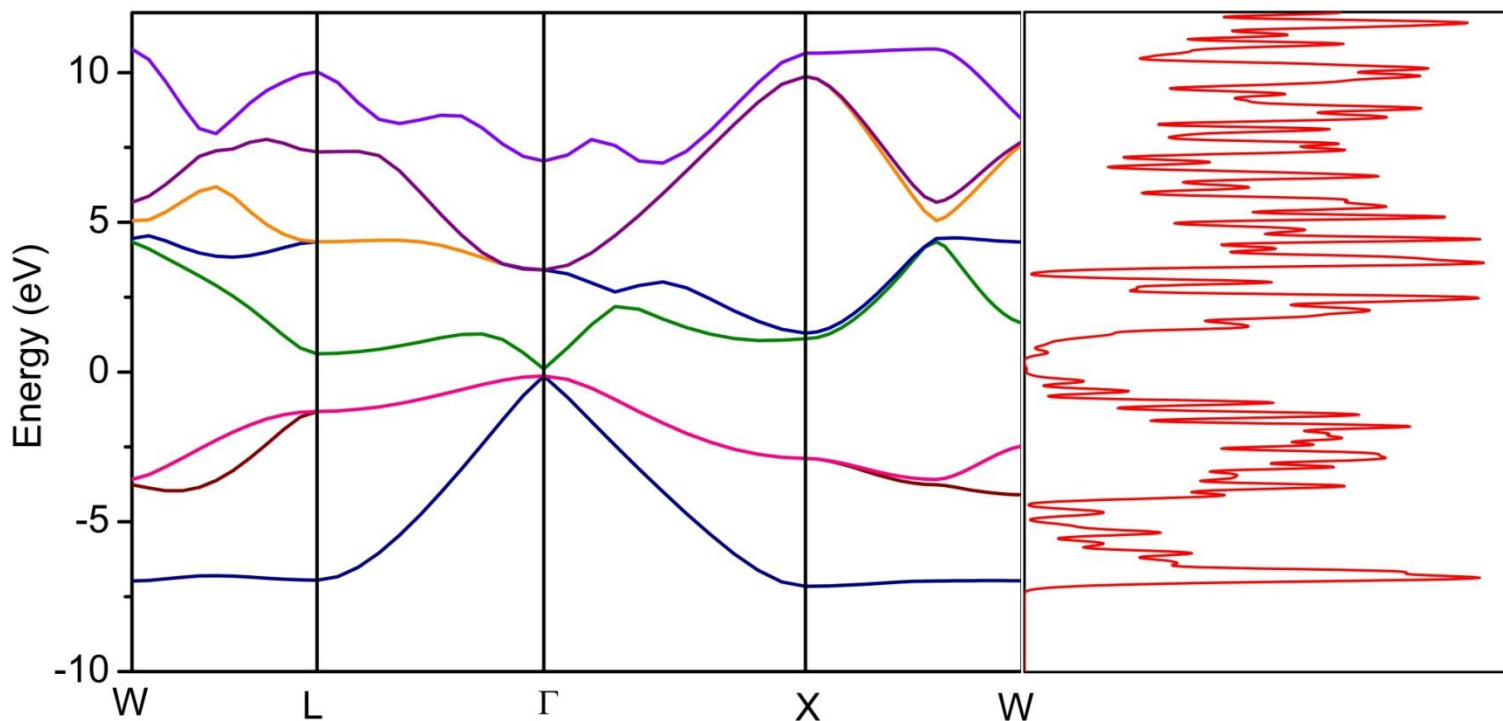
**N x y z 0 0 0 0**

**N x y z 0 0 0 0**

**...**

# 态密度计算

问题：态密度和能带有什么关系？



能带越密集的地方，态密度越大，能带越稀疏的地方，态密度越小？

不一定，但是大体规律是这样

# 态密度计算

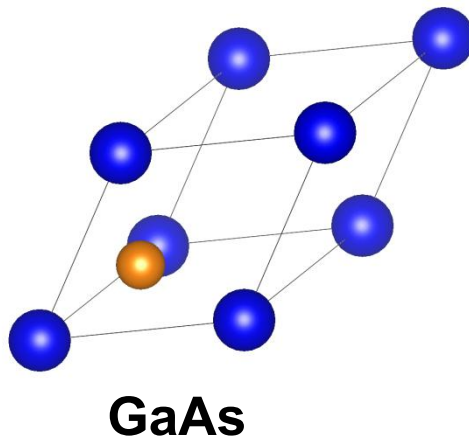
## 态密度计算步骤:

- 首先进行**SCF**计算
- 将**SCF**计算得到的**OUT.WG**拷贝成**IN.WG**，如果开了自旋还需把**OUT.WG\_2**拷贝成**IN.WG\_2**，此外，还需要拷贝**OUT.EIGEN**文件到**DOS** 计算目录下，并在**etot.input**中设置“**JOB=DOS**”以及“**IN.WG=T**”
- 数据处理，得到态密度图和画态密度的数据文件

# 态密度计算

进行**SCF**计算:

输入文件包括: **etot.input**, **atom.config**, **\*.UPF**



```
2  
Lattice vector  
3.9974000454      0.0000000000      0.0000000000  
1.9987000227      3.4618499884      0.0000000000  
1.9987000227      1.1539499961      3.2638634697  
Position, move_x, move_y, move_z  
33  0.250000000000  0.250000000000  0.250000000000  1  1  1  
31  0.000000000000  0.000000000000  0.000000000000  1  1  1
```

**atom.config**

```
4 1  
IN.ATOM = atom.config  
JOB =scf  
IN.PSP1 = Ga.SG15.NCPP.PBE.UPF  
IN.PSP2 = As.SG15.NCPP.PBE.UPF  
MP_n123 = 10 10 10 0 0 0
```

**etot.input**

# 态密度计算

进行DOS计算:

输入文件包括: **etot.input**, **atom.config**, \*.UPF

```
2  
Lattice vector  
 3.9974000454    0.0000000000    0.0000000000  
 1.9987000227    3.4618499884    0.0000000000  
 1.9987000227    1.1539499961    3.2638634697  
Position, move_x, move_y, move_z  
 33  0.250000000000  0.250000000000  0.250000000000  1  1  1  
 31  0.000000000000  0.000000000000  0.000000000000  1  1  1
```

**atom.config**

```
4 1  
IN.ATOM = atom.config  
JOB =DOS  
IN.PSP1 = Ga.SG15.NCPP.PBE.UPF  
IN.PSP2 = As.SG15.NCPP.PBE.UPF  
MP_n123 = 10 10 10 0 0 0  
IN.WG = T
```

**etot.input**

TIPS:

1. 计算态密度除了需要包含IN.WG文件以外, 还需要包含OUT.EIGEN文件不过这里的OUT.EIGEN不需要改名为IN.EIGEN保留原始文件名就可以了
2. 在使用HSE 计算态密度的时候, 还需要额外输入OUT.HSEWR\* 文件

# 态密度计算

数据处理：计算目录下运行plot\_DOS.py，运行完毕后会得到下列文件

```
dos.eps DOS.totalspin
```

eps格式的态密度图

画态密度所需的数据

不同轨道态密度

能量

总态密度

Energy	Total	Ga-d	Ga-s	Ga-p	As-s	As-p
-0.10734E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10712E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10690E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10668E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10646E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10624E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10602E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10579E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
-0.10557E+02	0.4780E-02	0.4742E-02	0.7657E-07	0.8852E-07	0.3775E-04	0.2972E-07
-0.10535E+02	0.1868E-01	0.1854E-01	0.8361E-06	0.3252E-06	0.1440E-03	0.2638E-06
-0.10513E+02	0.4083E-01	0.4052E-01	0.2102E-05	0.6944E-06	0.3109E-03	0.7389E-06
-0.10491E+02	0.7939E-01	0.7878E-01	0.4365E-05	0.1331E-05	0.5993E-03	0.1657E-05
-0.10469E+02	0.1416E+00	0.1405E+00	0.8150E-05	0.2347E-05	0.1061E-02	0.3270E-05
-0.10447E+02	0.2332E+00	0.2314E+00	0.1398E-04	0.3826E-05	0.1738E-02	0.5821E-05
-0.10425E+02	0.3615E+00	0.3588E+00	0.2218E-04	0.5781E-05	0.2633E-02	0.1113E-04
-0.10403E+02	0.5395E+00	0.5357E+00	0.3262E-04	0.8104E-05	0.3690E-02	0.2274E-04
-0.10381E+02	0.7430E+00	0.7381E+00	0.4451E-04	0.1053E-04	0.4792E-02	0.3874E-04
-0.10359E+02	0.9797E+00	0.9739E+00	0.5634E-04	0.1268E-04	0.5763E-02	0.6234E-04

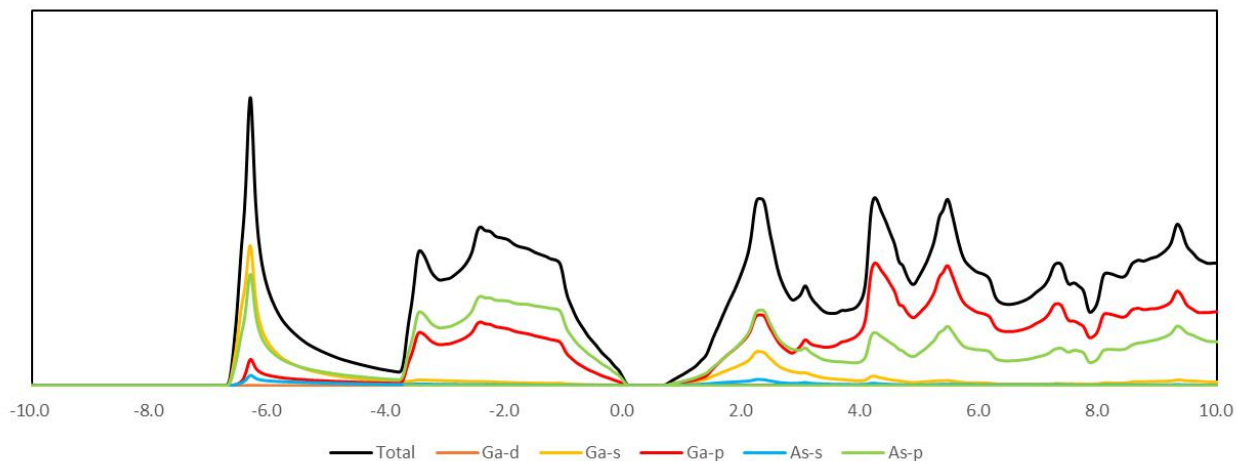
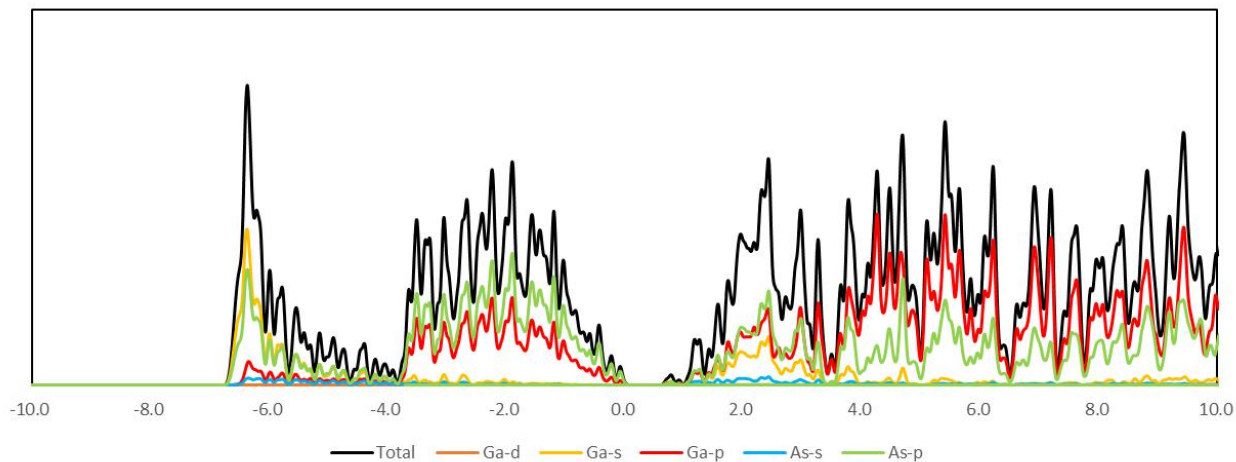
...

特别注意：如果开了自旋，还会有DOS.spinup和DOS.spindown

# 态密度计算

比较一下下面这两张图，哪张好看？

如何减小态密度中的毛刺？  
使用NONSCF加密K 或者DOS  
Interpolation



# 态密度计算

## 加密K点来平滑DOS曲线:

- 首先进行SCF计算
- 加密K点进行NONSCF计算（NONSCF计算的时候需要将OUT.VR以及OUT.VR\_2拷贝成IN.VR和IN.VR\_2，并在etot.input中设置“IN.VR=T”）
- NONSCF计算得到的OUT.WG拷贝成IN.WG，如果开了自旋还需把OUT.WG\_2拷贝成IN.WG\_2，并在etot.input中设置“JOB=DOS”以及“IN.WG=T”
- 数据处理，得到态密度图和画态密度的数据文件

```
4 1
IN.ATOM = atom.config
JOB =scf
IN.PSP1 = Ga.SG15.NCPP.PBE.UPF
IN.PSP2 = As.SG15.NCPP.PBE.UPF
MP_n123 = 10 10 10 0 0 0
```

SCF的输入文件etot.input

加密K点



```
4 1
IN.ATOM = atom.config
JOB =nonscf
IN.PSP1 = Ga.SG15.NCPP.PBE.UPF
IN.PSP2 = As.SG15.NCPP.PBE.UPF
MP_n123 = 15 15 15 0 0 0
IN.VR = T
```

NONSCF加密K点的输入文件etot.input

# 态密度计算

使用interpolation来平滑DOS曲线:

- 首先进行SCF计算
- 将SCF计算得到的OUT.WG拷贝成IN.WG，如果开了自旋还需把OUT.WG\_2拷贝成IN.WG\_2，并在etot.input中设置“JOB=DOS”以及“IN.WG=T”，此外还需要设置DOS\_DETAIL参数
- 数据处理，得到态密度图和画态密度的数据文件

# 态密度计算

interpolation方法参数设置:

```
4 1
IN.ATOM = atom.config
JOB = DOS
IN.PSP1 = Ga.SG15.NCPP.PBE.UPF
IN.PSP2 = As.SG15.NCPP.PBE.UPF
MP_n123 = 10 10 10 0 0 0
IN.WG = T
DOS_DETAIL = 1 10 10 10
```

使用如上输入文件

需要增加一行**DOS\_DETAIL**

**1** 代表需要使用**DOS\_interpolation** 默认值为**0**

**10 10 10** 代表计算**IN.WG** 时使用的**MP\_N123**

计算完成后, 写一个文件**DOS.input**

**0** 代表对所有原子做态密度分析

**1** 代表做**interpolation**

**0.05** 代表**interpolation**时**smear** 的能量

**5 5 5** 代表**interpolation**后的格子密度, 数字越大, 态密度越平滑, 计算也越慢

```
0
1
0.05
5 5 5
```

# 态密度计算

**interpolation**方法数据处理：计算目录下运行`plot_DOS_interp.x`，运行完毕后会也会得到下列文件

DOS.totalspin	columns: DOS_total,s,p,d,...
DOS.spinup	if spin=2
DOS.spindown	if spin=2
DOS.totalspin_projected	columns: DOS_total,s,pz,px,py,dz2,dxz,dyz,dxy,dx2-y2,...
DOS.spinup_projected	if spin=2
DOS.spindown_projected	if spin=2

**特别注意：**如果使用了**interpolation**方法也可以同时得到投影态密度需要的数据

# 有效质量计算

有效质量的定义:

$$\left(\frac{1}{m^*}\right)_{ij} = \frac{1}{\hbar^2} \frac{\partial^2 E_n(\bar{k})}{\partial k_i \partial k_j} \quad i, j = x, y, z$$

其中,  $x, y, z$ 是倒空间的方向,  $E_n(\bar{k})$ 是第 $n$ 个能带在 $\bar{k}$ 处的本征值

$$\frac{d^2 E}{dk^2} = \begin{pmatrix} \frac{d^2 E}{dk_x^2} & \frac{d^2 E}{dk_x k_y} & \frac{d^2 E}{dk_x k_z} \\ \cdot & \frac{d^2 E}{dk_y^2} & \frac{d^2 E}{dk_y k_z} \\ \cdot & \cdot & \frac{d^2 E}{dk_z^2} \end{pmatrix}$$

# 有效质量计算

EMC是使用有限差分法，而非能带拟合法，计算有效质量的小程序，目前它有FORTRAN和Python两个版本，目前它与CRYSTAL、VASP、CASTEP有接口，最近我们新增了EMC与PWmat的接口。

EMC的Python版本安装方法：

1. 从PWmat官网下载EMC程序
2. 下载之后解压，给emc.py文件添加可执行权限，具体命令为：  
`chmod +x emc.py`
3. 将emc.py所在的文件夹添加到环境变量

# 有效质量计算

## EMC的输入文件

```
.
0.000 0.000 0.000          ! K-POINT in the reciprocal crystal coord. (3 floats)↵
0.01                       ! step size in 1/Bohr units (1 float)↵
81                          ! band number, (1 integer)↵
V                           ! program identifier (1 char)↵
6.291999817  0.000000000  0.000000000 ! direct lattice vectors (3 floats)↵
0.755765092  7.652872670  0.000000000 ! direct lattice vectors (3 floats)↵
0.462692761  3.245907103 14.032346772 ! direct lattice vectors (3 floats)↵
```

- **band number.** For *PWmat* valence band number can be obtained as a half of the `NUM_ELECTRON` variable from the *REPORT* file (for non spin-polarized calculations).↵
- **program identifier:** V for *VASP* and *PWmat*.↵
- **direct lattice components** in *PWmat* in the `atom.config` under: `lattice`.↵

# 有效质量计算

## 3. Usage

1. Run SCF.
2. Generate k-point grid (in KPOINT file, you can move it to IN.KPT, and modify to IN.KPT format ):
  - Python version: `emc.py input_file(` FORTRAN version: `emc_gen`. Note: FORTRAN version requires input file to be named `inp`)
  - move KPOINT to IN.KPT (move KPOINT IN.KPT), then modify to IN.KPT format:

```
EMC 1.51py
19
Reciprocal
 0.0000000000    0.0000000000    0.0000000000  0.01
-0.0169868827    0.0000000000    0.0000000000  0.01
 0.0169868827    0.0000000000    0.0000000000  0.01
 0.0000000000   -0.0169868827    0.0000000000  0.01
 0.0000000000    0.0169868827    0.0000000000  0.01
 0.0000000000    0.0000000000   -0.0169868827  0.01
 0.0000000000    0.0000000000    0.0169868827  0.01
-0.0169868827   -0.0169868827    0.0000000000  0.01
 0.0169868827    0.0169868827    0.0000000000  0.01
 0.0169868827   -0.0169868827    0.0000000000  0.01
-0.0169868827    0.0169868827    0.0000000000  0.01
-0.0169868827    0.0000000000   -0.0169868827  0.01
 0.0169868827    0.0000000000    0.0169868827  0.01
 0.0169868827    0.0000000000   -0.0169868827  0.01
-0.0169868827    0.0000000000    0.0169868827  0.01
 0.0000000000   -0.0169868827   -0.0169868827  0.01
 0.0000000000    0.0169868827    0.0169868827  0.01
 0.0000000000    0.0169868827   -0.0169868827  0.01
 0.0000000000   -0.0169868827    0.0169868827  0.01
```

KPOINT

# 有效质量计算

## 3. Usage

```
19
2 0
0.0000000000 0.0000000000 0.0000000000 0.01
-0.0169868827 0.0000000000 0.0000000000 0.01
0.0169868827 0.0000000000 0.0000000000 0.01
0.0000000000 -0.0169868827 0.0000000000 0.01
0.0000000000 0.0169868827 0.0000000000 0.01
0.0000000000 0.0000000000 -0.0169868827 0.01
0.0000000000 0.0000000000 0.0169868827 0.01
-0.0169868827 -0.0169868827 0.0000000000 0.01
0.0169868827 0.0169868827 0.0000000000 0.01
0.0169868827 -0.0169868827 0.0000000000 0.01
-0.0169868827 0.0169868827 0.0000000000 0.01
-0.0169868827 0.0000000000 -0.0169868827 0.01
0.0169868827 0.0000000000 0.0169868827 0.01
0.0169868827 0.0000000000 -0.0169868827 0.01
-0.0169868827 0.0000000000 0.0169868827 0.01
0.0000000000 -0.0169868827 -0.0169868827 0.01
0.0000000000 0.0169868827 0.0169868827 0.01
0.0000000000 0.0169868827 -0.0169868827 0.01
0.0000000000 -0.0169868827 0.0169868827 0.01
```

IN.KPT

3.Run non-self consistent calculation using obtained k-point grid

4.Calculate effective masses and principal directions using eigenval file:

- get eigenval file: get\_eigenval.py
- Python version: emc.py input\_file eigenval (FORTRAN version: emc\_calc. Current folder should contain both inp and EIGENVAL files)